Project Title: Credit Card Fraud Detection

Problem Statement:

Credit card fraud is a significant problem for financial institutions and consumers, resulting in substantial financial losses and damage to customer trust. This project aims to develop a machine learning model to accurately identify fraudulent credit card transactions in real-time, minimizing financial losses and improving customer experience.

Data Source:

The project uses a publicly available dataset from Kaggle containing anonymized credit card transactions. The dataset includes features such as transaction amount, time, and various principal components derived from sensitive cardholder information, with a binary label indicating whether a transaction is fraudulent (1) or genuine (0).

Code and Model Explanations:

1. **Data Loading and Preprocessing:**
   - The code begins by loading the credit card transaction dataset using Pandas.
   - Data exploration is performed to understand the dataset's structure and characteristics.
   - Missing values are checked and handled.
   - Duplicate rows are identified and removed.
   - Outlier detection and handling is done.
   - The dataset is split into training and testing sets.
2. **Class Imbalance Handling:**
   - The SMOTE (Synthetic Minority Over-sampling Technique) is used to address the class imbalance problem, which is common in fraud detection datasets.
   - SMOTE creates synthetic samples of the minority class (fraudulent transactions) to balance the dataset and improve model performance.
3. **Model Training and Evaluation:**
   - Two models are trained and evaluated:
     - K-Nearest Neighbors (KNN)
     - Logistic Regression
   - Hyperparameter tuning is performed using GridSearchCV to find the best model settings.
   - Model performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC AUC score.
4. **Model Selection and Deployment:**
   - The model with the best performance is selected for deployment.
   - The selected model is saved using pickle for future use.

Instructions to Run the Code and Reproduce the Results:

1. **Environment Setup:**
   - Create a new Google Colab notebook.
   - Install necessary libraries:
2. **Data Loading:**
   - Upload the credit card transaction dataset (`creditcard.csv`) to your Google Drive.
   - Mount your Google Drive in the notebook:

3. **Execute Code Cells:**
    - Run the code cells in the notebook sequentially, following the order presented in the original code.
    - The code will perform data preprocessing, model training, evaluation, and selection.
4. **Reproduce Results:**
    - The final model and processed data are saved using pickle.
    - Load the saved model and data to reproduce the results:

Methodology:

- The dataset was preprocessed by handling missing values, removing duplicates, and addressing class imbalance using the SMOTE (Synthetic Minority Over-sampling Technique).
- The data was split into training and testing sets (80% and 20%, respectively).
- A K-Nearest Neighbors (KNN) model was employed for classification.
- Hyperparameter tuning was performed using GridSearchCV with 5-fold cross-validation, exploring different values for 'n_neighbors' (3, 5, 7, 9) and 'weights' ('uniform', 'distance').
- The model was evaluated using accuracy, F1-score, recall, and ROC AUC score.

Results:

- The optimized KNN model achieved the following performance metrics on the test set:
    - Accuracy: 0.9995
    - F1 Score: 0.9995
    - Recall: 1.0
    - ROC AUC Score: 0.9995

Interpretation:

- The model demonstrates exceptional performance in accurately classifying credit card transactions as fraudulent or non-fraudulent.
- The high recall score (1.0) indicates that the model correctly identifies all actual fraudulent transactions.
- The near-perfect accuracy, F1-score, and ROC AUC score further confirm the model's effectiveness.